

Ядро данных

Реплика amoCRM и шлюзы, через которые сервисы ходят в CRM.

- [Реплика amoCRM \(amo_db\)](#)
- [Как ходить в amoCRM из своего сервиса](#)
- [Вход в виджеты amoCRM](#)

Реплика amoCRM (amo_db)

Для разработчиков. Подробности устройства — в README репозитория `Buharestskie/amo_db`.

Что это

amo_db — копия нашего аккаунта amoCRM в обычной PostgreSQL. Сделки, контакты, задачи, примечания, неразобранное, каталоги, справочники и журнал событий. Изменение из amoCRM попадает в базу за несколько секунд.

Если сервису нужно **читать** данные amoCRM, он читает их отсюда, а не из API. У API лимит в 7 запросов в секунду на интеграцию и 50 на весь аккаунт — его делят все наши сервисы, виджеты и телефония, и когда он кончается, amoCRM сначала отвечает ошибками, а потом блокирует аккаунт. Из базы можно выбирать что угодно и сколько угодно.

Писать в amoCRM через базу нельзя: у всех ролей только чтение. Запись — как и раньше, через API amoCRM.

Как получить доступ

1. Попросить администратора завести роль с доступом «только чтение» к базе `amo_db`. Роли выдаются через панель pg_provisioner, пароль показывается один раз.
2. Попросить внести в список доступа внешний IP сервера, с которого будет подключение. Для своей машины разработчику достаточно запустить `./scripts/dev-db-access.sh` из репозитория amo_db — скрипт сам определит адрес. Если адрес часто меняется, есть туннель поверх SSH: `./scripts/db-tunnel.sh`.
3. Подключаться к `db.donula26.ru`, порт **6432**, база `amo_db`.

Порт 6432 — это пулер соединений. Он возвращает соединение в общий пул после каждой транзакции, поэтому не работают `LISTEN/NOTIFY`, временные таблицы и сессионные `SET`. Если приложение на asynsrg, кэш prepared statements нужно выключить (`statement_cache_size=0`), иначе под нагрузкой начнут плавать ошибки `prepared statement ... does not exist`.

Четыре правила чтения

Правило	Почему
Фильтровать <code>is_deleted = false</code>	Удалённое в amoCRM здесь не стирается, а помечается. Без фильтра в выборку попадут удалённые сделки и контакты.
В справочниках брать <code>is_current = true</code>	Статусы, воронки, поля, пользователи хранятся с историей версий: переименованный статус остаётся расшифровываемым в старых сделках. Для текущего названия нужна текущая версия.
Кастомные поля — из <code>amo_custom_field_values</code>	Там значения развёрнуты построчно, по ним удобно фильтровать. Разбирать JSONB в <code>amo_leads.custom_fields</code> не нужно.
Поиск контакта по телефону — через <code>amo_phone_tails(phones)</code>	<code>phone_normalized</code> — только первый телефон контакта. Поиск по любому номеру: <code>WHERE amo_phone_tails(phones) && ARRAY['9001112233']</code> , номер — последние 10 цифр.

Главные таблицы

Таблица	Что в ней
<code>amo_leads</code>	Сделки
<code>amo_contacts</code>	Контакты
<code>amo_tasks</code>	Задачи
<code>amo_notes</code>	Примечания: звонки, SMS, превью писем. Переписки из мессенджеров здесь нет
<code>amo_links</code>	Связи между сущностями — например, какие контакты у сделки
<code>amo_events</code>	Журнал изменений аккаунта: смены этапов, ответственных, удаления
<code>amo_statuses</code> , <code>amo_pipelines</code> , <code>amo_users</code> , <code>amo_custom_fields</code>	Справочники, которыми расшифровываются id в сделках

Все таблицы поле за полем — в `MODELS.md` в репозитории.

Чего в базе нет

- **Переписки из мессенджеров.** Схема под неё готова, наполнение отложено.
- **Компаний** отдельной таблицей. `company_id` в сделках и контактах есть.
- **Покупателей и сегментов** — в аккаунте они выключены.

Как понять, что данные свежие

`https://replika.donula26.ru/sync/state` показывает каждый поток синхронизации. Смотреть на `last_success_at` и `last_error`: если успех был в последние минуты и ошибки нет, поток работает. На `lag_seconds` не ориентироваться — у удалений и каталогов он часами растёт просто потому, что меняться нечему.

Если поток встал, мониторинг сам пишет в Telegram.

Как ходить в amoCRM из своего сервиса

Для разработчиков. Подробности устройства — в README репозитория `Buharestskie/amo_queue`

Коротко

Если ваш сервис **читает** данные amoCRM — читайте их из реплики, см. «Реплика amoCRM (amo_db)». Если сервис **пишет** в amoCRM или ему нужно то, чего в реплике нет, — ходите через шлюз `https://amo-queue.donula26.ru`, а не напрямую в amoCRM.

Зачем шлюз

amoCRM пускает не больше 7 запросов в секунду от одной интеграции и не больше 50 от всего аккаунта. Превышение — ошибка `429`, повторные превышения — блокировка всего аккаунта: встают все виджеты, дашборды и телефония разом.

Каждый сервис по отдельности может аккуратно держать свои 7 запросов, но про соседей он не знает — и вместе они пробивают общие 50. Шлюз ставит запросы всех сервисов в одну очередь и следит за обоими лимитами сразу.

Как подключиться

Поменять в конфиге сервиса один адрес — больше ничего:

Было	Стало
<code>https://donula.amocrm.ru/api/v4/leads?with=contacts</code>	<code>https://amo-queue.donula26.ru/api/v4/leads?with=contacts</code>

Пути, методы, тело запроса и заголовок `Authorization: Bearer <токен>` остаются прежними. Регистрировать сервис в шлюзе не нужно: он отличает интеграции по токену.

Обязательно передавайте имя сервиса в заголовке `X-Integration-Name`:

```
curl https://amo-queue.donula26.ru/api/v4/account \
-H "Authorization: Bearer $AMO_TOKEN" \
-H "X-Integration-Name: my-service"
```

Без имени в статистике и алертах виден только кусок хэша токена, и понять, чей это сервис, уже нельзя. Имя — латиница, цифры, точка, дефис, подчёркивание, до 40 символов.

Что может ответить шлюз

Почти всегда вы получаете ответ самой amoCRM без изменений. Свои ответы шлюз отдаёт в четырёх случаях:

Код	Что значит	Что делать
401	Нет заголовка <code>Authorization</code>	Передать токен
429 с телом <code>gateway queue is full</code>	Ваш сервис прислал больше, чем его очередь разгребает за 10 секунд	Повторить через <code>Retry-After</code> и растянуть отправку во времени. Это не amoCRM и не сбой шлюза — это ваш сервис шлёт пачкой
502	amoCRM недоступна	Повторить позже
504	amoCRM не ответила за 30 секунд	Повторить позже; если повторяется — упростить запрос

Сколько можно слать

На один токен — 6 запросов в секунду, на все сервисы вместе через шлюз — 30. Оставшиеся до 50 — запас для тех, кто в amoCRM ходит напрямую: реплика, телефония, сторонние виджеты.

Запросы сверх 6 в секунду не отбрасываются сразу, а ждут своей очереди до 10 секунд. Поэтому короткие всплески проходят незаметно, а массовые выгрузки — нет: их надо отправлять равномерно или разбивать на части с паузами.

Если фоновая выгрузка мешает быстрым запросам того же сервиса (кнопки в виджете начинают тормозить), заведите для фоновой задачи отдельную интеграцию в amoCRM со своим токеном — у неё будет своя очередь.

Лендинги ходят не сюда

Бэкенды квиз-лендингов (mfctest, bfl_fspp, do-nula-refinancing-2 и подобные) отправляют заявки не через шлюз, а через отдельный сервис `amocrm-service` — `https://donula26.ru/service/amocrm/`. Причины:

- **Токен amoCRM не лежит на лендинге.** Лендинг присылает готовую сделку, контакт и примечание, а токен знает только `amocrm-service`.
- **Защита от спама с форм.** Не больше 9 заявок с одного IP посетителя в час и 10 сущностей на один телефон в сутки. Поэтому в каждом запросе лендинг передаёт `ip` посетителя и его `phone`.
- **Заявка не теряется, если amoCRM тормозит.** `POST /leads/async` ставит цепочку «контакт → сделка → примечание» в очередь и сразу отвечает посетителю; дальше её доводит воркер.

Пускают туда только известные адреса. Новый лендинг — это новая строка в списке доступа сервиса: попросите администратора добавить его до запуска, иначе на каждую заявку будет ответ `403 Caller IP is not permitted`. Всё остальное — ручки, коды ответов, что делать при сбое — в README репозитория `Buharestskie/amocrm-service`.

Как посмотреть, что с вашим сервисом

- `https://amo-queue.donula26.ru/stats` — сейчас: темп, очередь и счётчики ответов по каждому сервису. Главное число — `queue_seconds`: сколько секунд ждёт запрос, пришедший прямо сейчас. Около нуля — всё хорошо.
- `https://amo-queue.donula26.ru/stats/history?hours=24` — то же по часам, до 7 суток назад.

Если у вашего сервиса растёт `queue_full` в счётчиках — он шлёт пачками, и часть запросов до amoCRM не доходит.

Вход в виджеты amoCRM

Для разработчиков виджетов. Подробности устройства — в README репозитория `Buharestskie/widget_auth`.

Зачем отдельный сервис

Виджет живёт внутри интерфейса amoCRM и своему бэкенду может предъявить только одно — токен `X-Auth-Token`, который amoCRM подписывает секретным ключом интеграции. Если проверять этот токен в каждом бэкенде, секрет интеграции окажется во всех виджетах сразу, и утечка из любого из них откроет все.

Поэтому секрет хранится в одном месте — в сервисе `widget_auth` на `https://widget-auth.donula26.ru`. Он проверяет токен amoCRM и выдаёт взамен свой, короткоживущий, подписанный отдельным секретом конкретного бэкенда. Бэкенд виджета секрета интеграции не видит никогда.

Сейчас так входят «Файлы сделки на Яндекс.Диске», чек-листы и шаблоны сообщений.

Как это выглядит со стороны виджета

1. Виджет вызывает обмен через `$authorizedAjax` — amoCRM сама подставит `X-Auth-Token`:

```
self.$authorizedAjax({
  url: 'https://widget-auth.donula26.ru/api/auth/exchange',
  method: 'POST'
}).done(function (response) {
  // response.access_token, response.expires_in (600 секунд)
});
```

2. Полученный `access_token` виджет передаёт своему бэкенду в заголовке `Authorization: Bearer <access_token>`.

3. Через 10 минут токен истекает — бэкенд ответит `401`, и виджет должен заново получить токен.

Как бэкенд проверяет токен

Токен — JWT, алгоритм HS256. Проверять обязательно все четыре вещи:

Что	Чем
Подпись	<code>ACCESS_TOKEN_SECRET</code> — свой у каждого бэкенда
Аудитория <code>aud</code>	<code>ACCESS_TOKEN_AUDIENCE</code> — своя у каждого бэкенда, например <code>yandex-widget-backend</code>
Издатель <code>iss</code>	<code>ACCESS_TOKEN_ISSUER=amocrm-widget-auth-service</code> — общий
Срок <code>exp</code> , <code>nbf</code>	Библиотека JWT проверяет сама; допуск на расхождение часов — несколько секунд

```
payload = jwt.decode(token, ACCESS_TOKEN_SECRET, algorithms=["HS256"],
                      audience=ACCESS_TOKEN_AUDIENCE, issuer=ACCESS_TOKEN_ISSUER, leeway=5)
```

Кто пришёл — в полях `account_id` и `user_id` (то же в `sub` как `"account_id:user_id"`). Это id пользователя amoCRM — по нему и решайте, что ему можно.

Не пропускайте проверку `aud`: именно она не даёт токену одного виджета пройти в бэкенд другого.

Подключить новый виджет

Делает администратор, у которого есть доступ к серверу:

1. В настройках интеграции amoCRM взять её ID (`client_uuid`) и секретный ключ.
2. Добавить виджет в список `widget_auth`: ID и секрет интеграции, новый случайный секрет бэкенда (от 32 символов) и уникальную аудиторию для него. Перезапустить сервис — список читается только при старте.
3. В настройки бэкенда виджета вписать тот же секрет бэкенда, ту же аудиторию и общий издатель.

Порядок команд и формат записи — в README репозитория, раздел «Подключить новый виджет».

Если вход не работает

Ответ	Что проверить
<code>401 Missing X-Auth-Token</code> от <code>widget_auth</code>	Запрос идёт не через <code>\$authorizedAjax</code> , а обычным <code>\$.ajax</code> или <code>fetch</code> — amoCRM подставляет токен только в первом случае
<code>401 Invalid X-Auth-Token</code> от <code>widget_auth</code>	Виджета нет в списке, неверный секрет интеграции или не совпала аудитория входящего токена. Обычно ошибка в только что добавленной записи
<code>401</code> от своего бэкенда	Токен истёк (прошло больше 10 минут) или секрет / аудитория в бэкенде не совпадают с записью в <code>widget_auth</code>
Ошибка CORS в консоли браузера	Виджет открыт не в нашем аккаунте amoCRM — обмен разрешён только со страниц <code>donu1a.amocrm.ru</code>